

USDC SDNY DOCUMENT ELECTRONICALLY FILED DOC #: _____ DATE FILED: <u>August 24, 2012</u>

UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

-----	X	
MICROSOFT CORPORATION,	:	
	:	11 Civ. 2365 (KBF)
Plaintiff,	:	
	:	<u>OPINION AND ORDER</u>
-v-	:	
	:	
DATATERN, INC.,	:	
	:	
Defendant.	:	
-----	X	
SAP AG and SAP AMERICA, INC.,	:	
	:	11 Civ. 2648 (KBF)
Plaintiffs,	:	
	:	
-v-	:	
	:	<u>OPINION AND ORDER</u>
DATATERN, INC.,	:	
	:	<u>(MARKMAN)</u>
Defendant.	:	
-----	X	

KATHERINE B. FORREST, District Judge:

In 2011, plaintiffs Microsoft Corporation ("Microsoft") and SAP AG and SAP America, Inc., ("SAP") (collectively, "plaintiffs") commenced separate lawsuits against DataTern, Inc., ("DataTern" or "defendant") for declarations of invalidity and non-infringement of two patents: U.S. Patent Nos. 6,101,502 (filed Sept. 25, 1998) ("`502 Patent") and 5,937,402 (filed June 19, 1997) ("`402 Patent"). Both patents-in-suit relate to inventions addressing object-oriented software applications interfacing with relational databases.

The '502 Patent is entitled "Object Model Mapping and Runtime Engine for Employing Relational Database with Object Oriented Software." The '402 Patent is entitled "System for Enabling Access to a Relational Database from an Object Oriented Program."

The parties are now before the Court on claim construction. With respect to the '502 Patent, they have requested construction of the following four terms¹:

1. "object model;"
2. "to create at least one interface object;"
3. "[at least one interface object] associated with an object corresponding to a class associated with the object oriented software application;" and
4. "runtime engine."

The parties have also requested that this Court construe five terms with respect to the '402 Patent:

1. "logical table;"
2. "logical primary key column;"
3. "designating one column of the logical table as the logical primary key column;"

¹ In terms of claims the parties are asking the Court to construe, there are certain differences between the "Joint Claim Construction Chart" (11 Civ. 2365, Dkt. No. 92, Ex. A ("JCCC")) submitted by the parties and the claims set forth in the PowerPoint presentations used at the Markman hearing. The Court has assumed that the latter documents prevail and that, to the extent a term was in an earlier document and not raised at the Markman hearing, it no longer requires construction. The parties shall promptly inform the Court if this assumption is incorrect.

4. "normalized relational schema object representing the logical table" (plaintiffs)/ "normalized relational schema object" (defendant); and
5. "generating."

The Court construes the above terms as set forth below.

I. LEGAL STANDARDS

A. Claim Construction

Claim construction is a question of law. Markman v. Westview Instruments, Inc., 52 F.3d 967, 979 (Fed. Cir. 1995), aff'd, 517 U.S. 370 (1996). This Court's task is neither to broaden nor narrow the claims – but rather to define them as a matter of law and thereby provide guidance as to the parameters of the patented invention. See Netword, LLC v. Centraal Corp., 242 F.3d 1347, 1352 (Fed. Cir. 2001).

In construing the meaning of a term, the question is not what that term would mean to an average lay person, but what that term would have meant to one "of ordinary skill in the art in question at the time of the invention." Phillips v. AWH Corp., 415 F.3d 1303, 1313 (Fed. Cir. 2005). The applicable date for claim construction purposes is the filing date of the first application to which the invention relates back – not the date upon which the patent issued. See Phillips, 415 F.3d at 1313.

There is a substantial body of law setting forth the appropriate tools with which a court should work in construing

claims, the order in which those tools should be utilized, and the weight that should be given to additional resources brought to bear on proffered constructions. A court may use intrinsic – and, if necessary, extrinsic – evidence. See Nazomi Commc'ns, Inc. v. Arm Holdings, PLC, 403 F.3d 1364, 1368 (Fed. Cir. 2005) (instructing courts to refer to intrinsic evidence first).

Intrinsic evidence includes the claims and specifications in the patent, as well as the patent's file history (or wrapper). See Datamize, LLC v. Plumtree Software, Inc., 417 F.3d 1342, 1348 (Fed. Cir. 2005). The single most important source for the meaning of a term is the language of the claim itself – the language of a claim defines the scope of the patent holder's exclusive rights. Phillips, 415 F.3d at 1312.

One skilled in the art is "deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." Phillips, 415 F.3d at 1313. The specification is the "single best guide to the meaning of a disputed term." Id. at 1315; see also On Demand Mach. Corp. v. Ingram Indus., Inc., 442 F.3d 1331, 1338, 1340 (Fed. Cir. 2006) ("[T]he scope and outer boundary of claims is set by the patentee's description of his invention," id. at 1338, and "the claims cannot be of broader scope than the invention that is set forth in the specification," id. at 1340.)

The patent itself provides critical guidance as to context. The Federal Circuit has instructed courts that even seemingly non-technical terms cannot be construed according to their "ordinary meaning" divorced from that context. See Tap Pharm. Prods., Inc. v. Owl Pharm., L.L.C., 419 F.3d 1346, 1354 (Fed. Cir. 2005). When referring to dictionary definitions, courts must therefore take care not to run afoul of this contextual principle. Phillips, 415 F.3d at 1322. Put another way, if the context of a patent suggests a more specialized or specific meaning, that context controls over the definition set forth in a dictionary (even a technical dictionary).

Patent prosecution history can be significant to proper claim construction. Statements the patentee may have made in connection with patent prosecution can be binding if the statements placed clear and unmistakable limits on claims. See Computer Docking Station Corp. v. Dell, Inc., 519 F.3d 1366, 1374-75; Phillips, 415 F.3d at 1317; see also Krippelz v. Ford Motor Co., 667 F.3d 1261, 1266-67 (Fed. Cir. 2012) (statements made during patent prosecution proceedings are binding on the patentee and can be considered as such during claim construction); Teleflex, Inc. v. Ficosa N. Am. Corp., 299 F.3d 1313, 1326 (Fed. Cir. 2002) ("[T]he prosecution history (or the file wrapper) limits the interpretation of claims so as to

exclude any interpretation that was disclaimed or disavowed during prosecution in order to obtain claim allowance.").

Binding patentees to statements they made during a patent prosecution prevents claim terms from varying as the need and situation changes. See Southwall Techs., 54 F.3d at 1578 ("A patentee may not proffer an interpretation for purposes of litigation that would alter the indisputable public record consisting of the claims, the specification and the prosecution history, and treat the claims as a 'nose of wax.' ").

B. Expert Witnesses

With respect to expert witnesses, courts apply Rule 702 of the Federal Rules of Evidence and evaluate whether the testimony is helpful. Ideally, the experts would testify live so that a court could also evaluate the credibility of the witness. Courts must also consider whether proffered opinions meet the standards set forth in Daubert v. Merrell Dow Pharmaceuticals, Inc., 509 U.S. 579 (1993). Rule 702 provides:

A witness who is qualified as an expert by knowledge, skill, experience, training, or education may testify in the form of an opinion or otherwise if: (a) the expert's scientific, technical, or other specialized knowledge will help the trier of fact to understand the evidence or to determine a fact in issue; (b) the testimony is based on sufficient facts or data; (c) the testimony is the product of reliable principles and methods; and (d) the expert has reliably applied the principles and methods to the facts of the case.

Fed. R. Evid. 702. In addition, a court should not consider mere ipse dixit of an expert – that is, conclusory statements without

analytical basis. Country Rd. Music, Inc v. MP3.com, Inc., 279 F. Supp. 2d 325, 330 (S.D.N.Y. 2003).

Plaintiffs and defendant submitted expert declarations in support of their proffered constructions. They declined to call either expert live at the Markman hearing or to cross-examine the other side's expert live at the hearing (though they were deposed). The Court was therefore unable to make credibility determinations, evaluate how the experts' proffered constructions withstood live cross-examination, or hear the experts' responses to the Court's specific inquiries. The Court must evaluate their opinions based on its overall view as to consistency between those opinions and the totality of the information available regarding the correct construction.

Both plaintiffs' and defendant's experts appear to have expertise in the area of object-oriented programming and relational databases. Neither side has questioned the qualifications of the other's expert or suggested that there are Daubert issues with respect to their proffered opinions.

The central issue for this Court in this Markman decision is to determine how one of ordinary skill in the art would interpret the contested claims. The temporal reference point is 1997-1998: the '402 Patent issued in 1999 based on an application dating back to 1997; the '502 Patent issued in 2000

based on two provisional applications dating back to 1997 (11 Civ. 2365, Dkt. No. 90, Exs. C, E) and an application in 1998.

Plaintiffs' expert, Dr. Anthony Hosking, has written a book on the state of the art in dynamic memory management for object storage in object-oriented programming languages. (11 Civ. 2365, Dkt. No. 91 ("Hosking Decl."), ¶ 4.) He has also written a number of articles on, inter alia, dynamic memory management for object storage, persistent storage for objects and concurrency in object-oriented systems. (Id.) Defendant submitted a declaration from Neeraj Gupta. (11 Civ. 2365, Dkt. No. 88, Ex. B ("Gupta Decl.").) Gupta received his Bachelor of Science in 1994 and his Masters of Science in Computer Engineering in 1995 from the Massachusetts Institute of Technology (Id. ¶ 2.) He does not list any publications but does state that he is a named inventor on five issued U.S. patents, including one relating to object-oriented programming. (Id. ¶ 6); see also U.S. Patent No. 6,629,153 (filed Sept. 17, 1997). In his declaration he states that he has a decade of experience in programming, development and design with respect to object-oriented applications and relational databases. (Id. ¶ 4.)

The Court carefully reviewed both declarations in connection with the other intrinsic and, as appropriate, extrinsic evidence necessary to construe the claims at issue.

II. THE CONSTRUCTIONS

A. The '502 Patent

1. "Object Model"

Plaintiffs' Proposed Construction: "A set of classes and the relationships among those classes underlying the implementation of the object-oriented software application." (11 Civ. 2365, Dkt. No. 92, Ex. A ("JCCC") at 1.)

Defendant's Proposed Construction: "No further construction is necessary. To the extent any further construction is deemed necessary: A template with a predetermined standardized structure that may include representing object classes, attributes of those object classes and inheritance relationships among classes." (Id.)

The correct construction of the term "object model" requires a brief description of how object-oriented programming relates to the inventions set forth in the '502 and '402 Patents. Object-oriented programming has been used by computer programming developers since the 1960s – with increasing popularity to today. (Hosking Decl. ¶ 13.) Some popular object-oriented programming languages include Java, C++ and Visual Basic. (Id.)

Object-oriented programming "typically organize[s] a software application into a collection of 'objects' that are defined by 'classes'." (Id. ¶ 15.) The "classes define the attributes of their objects" and how the objects relate to one another. (Id.)

Dr. Hosking's declaration quotes excerpts from various textbooks relating to object-oriented software applications applicable as of the 1997-1998 timeframe. (See id. ¶¶ 20-22.) These textbooks provide helpful information relating to how one of ordinary skill in the art would have read and understood the language of the patents and claims. These texts are consistent in their descriptions of object-oriented programming.

The parties have agreed that the proper construction of the term "object" is "an entity comprising attribute values (i.e., data) and methods (i.e., functionality) that can act on said attribute values." (11 Civ. 2365, Dkt. No. 92 at 2.) The parties have also agreed to a construction of the term "class" as "a definition that specifies attributes and behavior of objects, and from which objects can be instantiated." (Id.)

Hosking describes classes "as a template or cookie cutter from which individual objects are stamped out." (Hosking Decl. ¶ 25.) He quotes from the text Designing Object-Oriented Software as analogizing a class to a "factory": "You can think of a class as a template for a specific kind of object, or as a factory cranking out as many of its products as required." (Hosking Decl. ¶ 25 (quoting Rebecca Wirfs-Brock, et al., Designing Object-Oriented Software 22 (1990)).)

According to Hosking – and not contradicted by Gupta – object-oriented software applications are organized around a set

of classes, which "define the objects that will interact with one another." (Hosking Decl. ¶ 29.) Hosking states, and Gupta does not contradict, that this collection of classes "defines what is commonly referred to as the 'object model' of the application." (Id.)

The essential difference between the parties' proposed constructions with respect to the term "object model" reduces to the question of whether it "may include representing object classes" (defendant's proposed construction) or whether it must include a set of object classes (plaintiffs' proposed construction). Defendant's proposed construction relies upon the contention that the parties agreed to a construction of the term "classes" that requires the inclusion of both behaviors and attributes. (11 Civ. 2365, Dkt. No. 113 ("Def.'s Reply Br.") at 5.) Defendant argues that an embodiment of the invention depicted in Figure 3 of the specification has object models without behaviors, see '502 Patent fig.3. Thus, according to defendant, an object model may include object classes (that have both attributes and behaviors) – but does not necessarily have to.

In support of its proposed definition, defendant also points to the portion of the specification explaining Figure 3. See '502 Patent col.2 ll.40-48. This portion of the specification states: "The object model 14 is a template that

has a predetermined standardized structure. The illustrated object model includes attributes and inheritance relationships that are mapped to relational database features such as tables, rows, columns, keys, and foreign keys." Id.

In addition, defendant points to the first step of Claim 1 of the '502 Patent, which is "selecting an object model," id. col.7 l.56, arguing that this is expansive language (11 Civ. 2365, Dkt. No. 102 ("Def.'s Resp. Br.") at 23). According to defendant, plaintiffs' proposed construction would impose an external limitation by requiring that the object model in fact be connected to classes. (Id. at 24.)

The Court finds that while the language of the specification and Claim 1 do use the words as recited by defendant, those words do not directly answer the critical question of whether an object model must necessarily include classes. The Court therefore turns to extrinsic evidence.

Gupta states that: "A representation of the type of such an object is often referred to as a class (or the class of an object). Classes specify attributes and/or methods of a type of object, and may further specify relationships amongst object classes." (Gupta Decl. ¶ 11.) Gupta does not separately offer a definition "object model."

Hosking submits that based on texts extant during the relevant time period, the only way to understand an "object

model" is in connection to objects and classes. (Hosking Decl. ¶¶ 29-34.) Thus, there is no way to read in a possibility that an object model is not connected to or representing classes; it is mandatory, not discretionary. The Court agrees.

As plaintiffs note in their brief, "[e]ach asserted claim requires the selection of an object model and the mapping of that object model to schema in the relational database." (11 Civ. 2365, Dkt. No. 98 ("Pls.' Resp. Br.") at 18.) This mapping is central to the fundamental purpose of the patent:

"facilitate[ing] access to the relational database by object oriented software applications." '502 Patent at [57]. The specifications make clear that this process contemplates the mapping of class attributes to schema in relational databases. See id. col.2 ll.43-48. Therefore, to fulfill the central purpose of the patent, the object model must include classes.

While the parties have agreed to one construction of the term "class," that agreement does not prevent the Court from providing the correct construction of the phrase "object model" as used in the patents-in-suit. The intrinsic evidence suggests that classes need not always include behaviors. Indeed, as plaintiffs noted in the Markman hearing (see Markman Hr'g Tr. 49:8-50:8, July 30, 2012), Figure 3, which defendant cites as an embodiment of an object model that does not include behaviors, does include classes. '502 Patent fig.3. Thus, Figure 3 neither

serves as an example of an object model without classes nor supports the contention that the parties' stipulated definition of "classes" categorically requires them to include behaviors. Moreover, as the "Brief Summary of the Invention" states, "[t]he object model can be created from database schema or database schema can be created from the object model." '502 Patent col.11.56-58. There is no requirement that all database schema necessarily include both behaviors and attributes.

Plaintiffs also urge that the correct construction of "object model" must tie that phrase to the "object-oriented software application." (Pls.' Resp. Br. 20.) In support, they point to the fact that the patent's prior owner, FireStar, took a position in prior litigation with Red Hat that "object model" is a "set of classes, which map to a relational database, and the relationships among those classes." (Id. at 20-21.) However, at the Markman hearing, defendant represented that the FireStar/Red Hat litigation never went as far as claim construction. (See Hr'g Tr. 26:13-27:2.)

The context of the patent is critical to whether the inclusion of the additional language is necessary: the invention requires an object-oriented software application. Accordingly, it is appropriate to include that language in the construction of "object model." Support for this is found in the '502 Patent Abstract: "The Database schema, object model and mapping are

employed to provide interface objects that are utilized by a runtime engine to facilitate access to the relational database by object oriented software applications." '502 Patent at [57]. Similar language is repeated in the "Brief Summary of the Invention." '502 Patent col.1 ll.59-62. In addition, Figure 1 shows an object model being used in connection with an object-oriented application. Id. fig.1.

Defendant points out that the specification also recites that "[t]he object model 14 is a template that has a predetermined standardized structure." Id. col.2 ll.39-40. Plaintiffs argue that, read literally, this type of "template" could capture Microsoft Excel spreadsheets or other "templates." (Pls.' Resp. Br. 21.) Again, the context of the patent itself limits the standardized template to one involving object-oriented software applications. Accordingly, the correct construction of "object model" is the following: a template with a predetermined standardized structure both relating to an object-oriented software application and including object classes and inheritance relationships among classes.

2. "To Create At Least One Interface Object"

Plaintiffs' Proposed Construction: "To generate code for at least one class and instantiate an object from that class, where the object is not part of or generated by the object oriented application and is used to access the database." (JCCC 6.)

Defendant's Proposed Construction: "No further construction is necessary. To the extent any further

construction is necessary: 'to create' means to make. 'Interface object' means 'An object by which the object oriented application accesses the relational database via a runtime engine.'" (Id.)

Defendant does not believe this phrase needs additional construction. (Id.) Plaintiffs urge that construction is needed both "to help the jury understand its meaning within the larger claim context" and also to effect the disavowal of scope that defendant made in a statement during patent prosecution. (11 Civ. 2365, Dkt. No. 89 ("Pls.' Opening Br.") at 11.) The language at issue appears in the third step of Claim 1 ("employing the map to create at least one interface object," '502 Patent col.7 l.59) – and, according to plaintiffs, has parallel language in Claim 10 ("code generator that employs said map to create at least one interface object," id. col.8 ll.32-33). (Pls.' Opening Br. 12.)

The parties do agree that to "create at least one interface object" requires some form of creation the end result of which is an "interface object." The parties do not agree on the method of the creation. Plaintiffs argue that "creation" as used here necessarily requires code generation (id. at 13); defendant argues in contrast that "creation" simply means "to make" and does not necessarily require code generation (11 Civ. 2365, Dkt. No. 88 ("Def.'s Opening Br.") at 27-29).

The language of Claim 1 itself refers to "employing the map" to create at least one interface object. '502 Patent col.7

11.59-60. Figure 1 of the specification shows the object model going directly into the mapping tool, which then goes directly to the map, which then goes directly into the code generator. Id. fig.1. However, Figure 1 does allow for the map to bypass the code generator and go directly into the runtime engine. Id. Nevertheless, as discussed at the Markman hearing, despite the fact that the runtime engine has a two-way arrow into and out of the interface objects, '502 Patent fig.1, interface objects only come out of the code generator. (See Hr'g Tr. 71:2-72:25.) Figure 1 therefore requires code generation to be a necessary part of the creation of interface objects. While Figure 1 is only one embodiment of the invention, defendant's own expert conceded that he is not aware of any embodiments of the '502 Patent that do not require code generation. (See 11 Civ. 2365, Dkt. No. 117, Ex. A ("Gupta Depo. Tr.") at 199:15-21.)

Defendant argues that the doctrine of claim differentiation requires this Court to construe the language as not requiring code generation. (Def.'s Opening Br. 29.) In contrast to the language in Claim 1, the language in Claim 10 specifically refers to "a code generator that employs said map to create at least one interface object associated with an object corresponding to a class associated with the object oriented software application." '502 Patent col.8 11.32-35. However, there are other language differences between Claims 1 and 10

that further differentiate them. For instance, Claim 1 refers to a "method," id. col.7 l.33, and Claim 10 refers to a "computer program fixed on a computer-readable medium," id. col.8 ll.24-25. These and other language differences between Claims 1 and 10 mean that construing the words "to create" to require code generation will not render the scope of the claims the same.

The second dispute relating to construction of this phrase is whether the "interface object" can be part of or generated by the object oriented application. During the reexamination process, defendant disclaimed that interface objects were part of or generated by the object-oriented application. In a supplemental reply submitted in that proceeding, defendant stated: "The [prior art] user objects are not to be confused with the interface objects recited in claim 1 of the present patent, which interface objects are not part of nor generated by the object oriented software application." (11 Civ. 2365, Dkt. No. 90, Ex. M at 10-11.) This disclaimer is binding.

Accordingly, the Court adopts plaintiffs' construction.

3. [At Least One Interface Object] "Associated with an Object Corresponding to a Class Associated with the Object-Oriented Software Application"

Plaintiffs' Proposed Construction: "Where an interface object class is generated for each object model class, to define methods callable by an instance of the object model class." (JCCC 10-11.)

Defendant's Proposed Construction: "No further construction is necessary. To the extent any further

definition is necessary, 'an object' and 'a class' have their agreed upon definitions. To the extent any further definition is necessary, the phrase means: 'The interface object (as defined above) corresponds to an object instantiated from a [single] class associated with the object-oriented software.'" (Id.)

Plaintiffs' primary assertions with respect to this phrase are: (1) that the dual use of the phrase "associated with" will be confusing to the jury, and (2) that the term "associated" is used ambiguously and requires construction. (Pls.' Opening Br. 17.) Defendant and this Court disagree.

Plaintiffs' proposed construction requires that there be a one to one relationship between "interface object class" and "object model class" – but there is no basis for this requirement in the specification or claim language.

According to plaintiffs, the phrase "associated with" should also be construed in order to define the method of achieving the claimed association. (See Pls.' Resp. Br. 32 (citing Akamai Techs., Inc. v. Limelight Networks, Inc., 629 F.3d 1311, 1327 (Fed. Cir. 2010)).)

At the Markman hearing, plaintiffs agreed to defendant's construction with one modification: "the interface object (as defined above) corresponds to an object instantiated from a **single** class associated with the object oriented software." (See Hr'g Tr. 83:13-85:2 (emphasis added).) This construction achieves plaintiffs' expressed desire to eliminate dual and potentially confusing usage of the phrase "associated with" and

also achieves further definition of the manner of association by referring to "the interface object" that "corresponds." This construction is also supported by the specification. See '502 Patent col.2 ll.13-24 (referring to Figures 1-7); col.2 ll.28-48 (describing Figure 1); col.6 ll.8-64 (describing Figures 5, 6, and 7). The Court therefore adopts this construction.

4. "Runtime Engine"

Plaintiffs' Proposed Construction: "Software that (i) the object oriented software application depends on to run, (ii) must be running to execute the object oriented software application, (iii) uses the map in its processing, and (iv) is not part of the object oriented software application." (JCCC 14.)

Defendant's Proposed Construction: "Software, which is not directly part of the object-oriented application, that the object-oriented application uses to access the relational database." (Id.)

Software is necessary to enable the object-oriented application to interface with the relational database. The service that does this is the runtime engine. (See Hosking Decl. ¶¶ 91-92.) Both parties agree that runtime engine should be distinguished from runtime environment and runtime library. (See Def.'s Reply Br. 9 ("[T]he examiner allowed the claims because the 'runtime engine' was properly distinguished from 'runtime environment' and 'runtime library.'")) Defendant argues that their construction captures this distinction (see id.); plaintiffs strongly disagree.

According to plaintiffs, their proposed construction carefully "mirrors language that DataTern used to distinguish its invention from the prior art and is also consistent with the other intrinsic evidence." (Pls.' Opening Br. 20.) This Court agrees.

First, during the reexamination, the asserted claims had been rejected based on prior art. (See, e.g., 11 Civ. 2365, Dkt. No. 90, Ex. N at 15.) Defendant submitted an affidavit from Mark Eisner, the Chief Technology Officer of FireStar, its sister company. In particular, the Eisner affidavit stated that the runtime engine was not the "runtime environment" referred to in the prior art reference. (See 11 Civ. 2365, Dkt. No. 88, Ex. D-3 at 3.) Eisner also stated that a runtime engine "connotes a service that delivers specific functionality that will enable an application to run," (id. at 4), and that it "operates during runtime execution of the object-oriented application," (id. at 6). The examiner agreed with the distinction Eisner put forward and provided a definition of "runtime-engine" from "PCMAG.COM."² (See 11 Civ. 2365, Dkt. No. 90, Ex. O at 2.) This definition states that runtime engine is "[s]oftware that certain applications depend on to run in the computer" and that "must be

² At the Markman hearing, the Court inquired as to whether it was appropriate to reference a definition published more recently than the timeframe of 1997-1998. No evidence was put forward that the definition changed in any way between 1997 and 1998 and the date that definition was published in PCMAG.COM. The parties did not suggest that there was a difference. (See, e.g., Hr'g Tr. 94:21-97:1, 100:13-102:10.)

running . . . to execute" the software application. (11 Civ. 2365, Dkt. No. 90, Ex. P.) The Examiner made this definition part of the file wrapper and therefore intrinsic evidence.

Accordingly, the Court adopts plaintiffs' proposed construction.

B. The '402 Patent

1. "Logical Table"

Plaintiffs' Proposed Construction: "Table outside a relational database that can be utilized like a physical table and that results from a normalization process that divides at least one of the physical tables into two or more tables to minimize duplication." (JCCC 25.)

Defendant's Proposed Construction: "Representation, which exists outside the relational database, of one or more columns of one or more physical tables or views." (Id.)

The Abstract of the '402 Patent states that:

"Logical tables and logical keys are employed to facilitate interaction between user applications and a relational database. Each logical table is a group of at least one column from a table or view associated with a relational database, and can be utilized like a relational table or view. . . . Once a denormalized table is captured from a database the normalization process allows the user to define difference logical tables using subsets of the columns of the table."

'402 Patent at [57].

There are two areas of dispute between the parties regarding their proposed constructions for "logical table." The first is whether the logical table is more than merely a "representation"; the second is whether a logical table results from a normalization process.

The specification clearly contemplates that logical tables are "tables" and not representations of tables. They can be used like physical tables. For instance, the Abstract of the '402 Patent notes that "logical tables interact with the mapping process in the same manner as physical tables." '402 Patent at 57]. Similarly, the Brief Summary of the Invention describes "[e]ach logical table" as "a group of at least one column from a physical table or view associated with a relational database, and can be utilized like a physical table." '402 Patent col.1 11.47-50. The specification also provides that: "These logical tables . . . are treated by the mapping process . . . exactly as if they are real physical tables in the database." Id. col.4 11.39-41.

The specification also supports plaintiffs' contention that logical tables result from a "normalization process that divides at least one of the physical tables into two or more tables to minimize duplication." (JCCC 25.) For instance, the Abstract states that "[o]nce a denormalized table is captured from a database the normalization process allows the user to define different logical tables using subsets of the columns of the table." Id. at [57]. At column 4, lines 29-34, the '402 Patent specification states: "[T]he normalization process allows the user to define different tables using subsets of columns AP. This new table is hence a subset of a real physical table in the

database and is henceforth referred to as a logical table." The Brief Summary of the Invention contains similar language. See id. col.1 ll.54-57.

Accordingly, the Court adopts plaintiffs' proposed construction for this term.

2. "Logical Primary Key Column"

Plaintiffs' Proposed Construction: "Column of the logical table with unique values in each row." (JCCC 29.)

Defendant's Proposed Construction: "No further construction is necessary. To the extent that further construction is deemed necessary: 'Column of logical table associated with a logical primary key value, i.e., a unique identifier of a row in a logical table.'" (Id. at 29-30.)

Defendant urges that no construction for "logical primary key column" is necessary. However, it is clear that the term is not self-evident based on the differing constructions proposed by the parties.

The disagreement between the parties is obscured by their proposed language. Plaintiffs suggest that the correct construction is a "column of a logical table with **unique** values in **each** row." (Id. at 29 (emphasis added).) In contrast, defendant's proposed language only requires that a column of a logical table be **associated with** a value, rather than that it contain key values. Defendant argues that plaintiffs "confuse[] logical tables with physical tables in a relational database." (Def.'s Resp. Br. 14.) According to defendant, because physical

tables are designed to store data, and logical tables to access data in a relational database, it need not be the case that a logical table must contain data rather than simply be associated with it (as defendant proposes in its construction). (See id.) Defendant's argument has initial appeal but is contradicted by the language of the specification.

For instance, the specification states that the "integrator", an element of the invention not here at issue, "guarantees that the user-defined logical table and keys will behave exactly as if they were real tables and keys in the relational database." '402 Patent col.5 l.67-col.6 l.2. It also states that "[t]he relational database ensures that the source table column contains a value that matches exactly with any one primary key value in the target table. Thus a relationship between the objects from the two tables is established." Id. col.9 ll.27-31.

In the DataTern and Staples litigation, see DataTern, Inc. v. Staples, Inc., No. 2:10-cv-133-TJW-CE (E.D. Tex.) (filed Apr. 19, 2010), defendant argued for the construction that plaintiffs now propose. (See 11 Civ. 2365, Dkt. No. 90, Ex. K at 4.) Accordingly, the Court finds no support for defendant's proposed construction and adopts plaintiffs'.

3. "Designating One Column of the Logical Table as a Logical Primary Key Column"

Plaintiffs' Proposed Construction: "Selecting by the user a single column of the logical table as a new primary key column." (JCCC 32.)

Defendant's Proposed Construction: "'Logical table' has the meaning identified above. 'Logical primary key column' has the meaning identified above. . . . No further construction is necessary. To the extent that any further construction is deemed necessary, the following constructions apply: 'Designating' has meaning to one of ordinary skill in the art having read the '402 patent and prosecution history and there is no limitation in the claim as to how it is to be selected." (Id. at 32-33.)

The difference between plaintiffs' and defendant's proposed constructions is the role of the "user." According to plaintiffs, "the user select[s] a single column as the primary key column, as opposed to, for example, automatically carrying such a designation from the relational database when a logical table is created through normalization." (Pls.' Opening Br. 37.)

Defendant points out that Claim 1 does not specify that "the user" plays any role in designating one column of the logical table as the logical primary key column. (Def.'s Resp. Br. 15.) Defendant argues that to import this limitation into the claim is to create something out of thin air – which prevailing patent law clearly does not allow. (Id.) While there is some support for plaintiffs' construction, it is weak, and the Court agrees that the absence of any reference to "the user" in the claim language itself controls. See Amgen Inc. v. Hoechst Marion Roussel, Inc., 314 F.3d 1313, 1325 (Fed. Cir. 2003) (discussing the danger of improperly importing limitations to a

claim). Claim 13, for instance, refers to Claim 8 with the addition of the limitation of "user defined." See Tandon Corp. v. U.S. Int'l Trade Comm'n, 831 F.2d 1017, 1023 (Fed. Cir. 1987) (noting that the use of different words or phrases in separate claims leads to a presumption of a difference in meaning and scope).

Plaintiffs assert that their construction is supported by the portion of the specification stating "[t]he user must define new constraints for the new logical tables even if they are the same as the constraints on the physical table." '402 Patent col.5 ll.8-10. It is not at all clear, as plaintiffs assert, that this language is intended to limit the designation to "user defined." The term "constraints," as used here, would need to be defined before that assumption could be made, and neither of the parties have requested such a construction.

Accordingly, the Court finds that the plain wording of the Claim does not require the user to play a role in the designation step. The Court therefore adopts defendant's proposed construction for this term.

4. "Normalized Relational Schema Object Representing the Logical Table"

Plaintiffs' Proposed Construction: "An object containing schema of the logical table that resulted from a normalization process that divided the physical table into two or more tables to minimize duplication." (JCCC 34.)

Defendant's Proposed Construction: "A construct representing a logical table, or portion thereof, in which a logical primary key column has been designated." (Id. at 34, 40.)

Plaintiffs' and defendant's proposed constructions for this phrase differ in terms of whether the relational schema object must undergo a normalization process. Defendant argues that a normalization process is not necessary, and plaintiffs argue it is. Claim 1 recites the step of "forming a normalized relational schema object." '402 Patent col.15 l.63. Defendant argues that while somehow a "normalized relational schema object" comes into existence, the claim does not limit how that process occurs. (See Def.'s Resp. Br. 17.) According to defendant, a "normalized relational schema object" could be formed directly from the capture process without intervening normalization. (See id. at 17-18.) As support for this, defendant points to Claim 2 which recites an additional step of "normalization." (Id. at 18); see also '402 Patent col.16 ll.7-8 ("2. The method of claim 1 including the further step of employing normalization to create foreign key relationships"). Defendant is correct that a normalization step is part of Claim 2, but ignores that Claim 2 is explicitly about the creation of foreign key relationships and does not relate explicitly to "forming a normalized relational schema object." Id. col.16 ll.7-11. Therefore, referring to Claim 2 is not comparing apples to apples.

Defendant also points to Claim 17 which recites a "system for enabling access to a relational database from an object oriented program." '402 Patent col.17 ll.41-42. Claim 17 refers to a "normalization process for inputting one or more denormalized relational schema objects," id. col.17 ll.43-44, and then "forming a normalized schema object, responsive to said set one or more denormalized relational schema objects," id. col.17 ll.47-49. Thus, according to defendant, if a normalization process was always needed, it would be recited in Claim 1. (Def.'s Resp. Br. 17.) Claim 17 is not expanding upon Claim 1 (nor does defendant contend that it does). It is instead claiming a "system." While defendant is correct that this claim explicitly refers to a "normalization process", '402 Patent col.17 l.43, that reference does not require the concept of normalization to be left out of the "forming" step in Claim 1, id. col.15 l.63. The context of the patent is critical here. The language of the specification and the context of the overall patent suggests no other normalization procedure. Indeed, they suggest that a normalization process is always a part of the invention. For instance, the Abstract states that "[a] normalization process allows creation of integrator relational schema objects from existing captured tables." '402 Patent at [57]. Figure 5 of the specification shows a capture process that then goes either into a normalization process or directly into

relational schema objects. Id. fig.5. But there is a two-way arrow from the relational schema objects box into the normalization process. Id. This suggests that the normalization process is always, in this embodiment, part of the "forming" step for relational schema objects. No other embodiment is suggested by the patent or defendant's expert.

For the second portion of their proposed construction, plaintiffs refer to Gupta's deposition in which he states that normalization is "typically used to separate data into more than one table to help minimize duplication." (Pls.' Resp. Br. 9 (quoting Gupta Depo. Tr. 62:17-63:8).)

The Court adopts plaintiffs' proposed construction.

5. "Generating"

Plaintiffs' Proposed Construction: "Creating without user input." (JCCC 44.)

Defendant's Proposed Construction: "No further construction is necessary. To the extent any further construction is necessary: 'Generating' means 'creating or producing.'" (Id.)

Plaintiffs and defendant disagree as to whether, if further construction is necessary, "generating" may or may not include user input. Plaintiffs correctly note that during the prosecution history of the '402 Patent, defendant specifically disclaimed user input in order to avoid prior art. During reexamination proceedings, defendant argued that referenced prior art was distinguishable because it disclosed defining

classes by user input, or "allow[ing] the user to define" or having "classes responsive to user input manually keyed in by the user." (11 Civ. 2365, Dkt. No. 90, Ex. X at 14.) As a result of this disclaimer, defendant is limited to a definition of generation that does not involve user input.

Accordingly, plaintiffs' proposed construction is adopted for this term.

CONCLUSION

The claims at issue are construed as set forth above. For purposes of clarity, the Court provides the construction for each of the disputed terms (except those that are the subject of a separate summary judgment motion) below:

<u>Term</u>	<u>Construction</u>
Object Model	A template with a predetermined standardized structure both relating to an object-oriented software application and including object classes and inheritance relationships among classes.
To Create at Least One Interface Object	To generate code for at least one class and instantiate an object from that class, where the object is not part of or generated by the object oriented application and is used to access the database.
[At Least One Interface Object] Associated with an Object Corresponding to a Class Associated with the Object Oriented Software Application	The interface object corresponds to an object instantiated from a single class associated with the object oriented software.

Runtime Engine	Software that (i) the object oriented software application depends on to run, (ii) must be running to execute the object oriented software application, (iii) uses the map in its processing, and (iv) is not part of the object oriented software application.
Logical Table	Table outside a relational database that can be utilized like a physical table and that results from a normalization process that divides at least one of the physical tables into two or more tables to minimize duplication.
Logical Primary Key Column	Column of the logical table with unique values in each row.
Designating One Column of the Logical Table as the Logical Primary Key Column	"Logical table" has the meaning identified above in this opinion. "Logical primary key column" has the meaning identified above in this opinion. No further construction is necessary.
Normalized Relational Schema Object Representing the Logical Table (plaintiffs)/ Normalized Relational Schema Object (defendant)	An object containing schema of the logical table that resulted from a normalization process that divided the physical table into two or more tables to minimize duplication
Generating	Creating without user input.

SO ORDERED:

Dated: New York, New York
August 24, 2012

Katherine B. Forrest

KATHERINE B. FORREST
United States District Judge